### **Abstract**

This report presents a project that applies multiple deep learning architectures to the Music Genre Classification task and chooses DenseNet architecture based on the best accuracy and classification metrics. DenseNet is known for its efficiency in computation and its ability to handle compact models effectively. The project focused on fine tuning hyperparameters of a DenseNet-201 model to process mel spectrograms of different music genres, aiming to achieve high classification accuracy. Key achievements include optimizing the model’s parameters and evaluating its performance against a benchmark dataset. The results highlight substantial improvements in classification metrics, showcasing the model’s robustness and practical potential.

### **Introduction & Problem Statement**

Image classification is a fundamental aspect of computer vision with applications ranging from healthcare to autonomous vehicles. Our project focused on mapping the genre classification problem on image classification by extracting mel spectrograms of music audios. This project addresses these challenges by utilizing the DenseNet-201 architecture, known for its dense layer connections that enhance feature reuse and gradient flow.

The project’s main objectives were:

* To evaluate DenseNet-201 with multiple hyperparameters.
* To optimize the model’s performance.
* To compare its results with standard benchmarks.

### **Literature Review**

DenseNet, introduced by Huang and colleagues, revolutionized convolutional neural networks with its dense connectivity pattern. Unlike traditional models that connect layers sequentially, DenseNet links all layers directly, ensuring better gradient flow and feature reuse. Research shows that DenseNet often outperforms legacy models like ResNet and VGGNet, especially in scenarios requiring compact, efficient models. This project builds on these findings to explore DenseNet-201’s practical applications in image classification.

### **Methodology**

The project followed a systematic approach:

1. **Data Preprocessing**:  
   * Gathering and cleaning the dataset.
   * Applying data augmentation techniques to improve generalization.
2. **Model Architecture**:  
   * Adapting DenseNet-201 with unfreezing layers for the dataset’s requirements.
   * Modifying the output layer to match the classification task.
3. **Training**:  
   * Using pre-trained weights as a starting point.
   * Optimizing hyperparameters such as learning rate, batch size, dropout, regularization, number of layers unfreezed et.
4. **Evaluation**:  
   * Employing metrics like accuracy, precision, recall, and F1-score.
   * Testing the model’s generalization on unseen data.

### **Implementation Details**

* **Frameworks and Tools**: TensorFlow, Keras, and Python for data handling and visualization.
* **Model**: Fine-tuned DenseNet-201 architecture.
* **Hardware**: RTX-A4000
* **Training Process**: The model was trained over 100 epochs with a batch size of 32. A learning rate scheduler helped maintain adaptive learning throughout. We unfreezed last 50 layers of the model to fine tune it on our dataset.
* **Custom Code**: Scripts were developed for data loading, preprocessing, and training.

### **Results & Discussion**

The DenseNet-201 model achieved:

* Training accuracy of 91%.
* Validation accuracy of 81%.
* Test accuracy of 83.64%.

Visualization tools such as confusion matrices and classification reports demonstrated the model’s strong performance in distinguishing between different classes. Compared to other baseline models, DenseNet-201 showed clear advantages in feature propagation and compact design.

Some challenges included overfitting on smaller datasets and computational demands during training. Future work could focus on advanced regularization methods and hardware optimization to address these issues.

### **Conclusion & Future Work**

This project successfully showcased DenseNet-201’s capabilities in image classification, balancing high accuracy with computational efficiency. While the results are encouraging, there is room for further exploration. Future work could involve:

* Using larger and more diverse datasets to enhance generalization.
* Testing hybrid models that combine DenseNet with other architectures.
* Optimizing the model for real-time deployment.

The outcomes of this study provide a solid foundation for utilizing DenseNet-201 in practical and scalable solutions across a wide range of industries.